By Dariusz Dwornikowski
Published: 2006-08-24 12:02

LDAP Authentication In Linux

This howto will show you howto store your users in LDAP and authenticate some of the services against it. I will not show howto install particular packages, as it is distribution/system dependant. I will focus on "pure" configuration of all componenets needed to have LDAP authentication/storage of users. The howto assumes somehow, that you are migrating from a regular passwd/shadow authentication, but it is also suitable for people who do it from scratch.

Requirements

- **OpenLDAP**
- **pam_ldap**
- **nss_ldap**
- **PADL migrationtools**

Introducion

The thing we want to achieve is to have our users stored in LDAP, authenticated against LDAP (direct or pam) and have some tool to manage this in a human understandable way.

This way we can use all software, which has ldap support or fallback to PAM ldap module, which will act as a PAM->LDAP gateway.

More information on LDAP idea can be found on Wikipedia: **LDAP wikipedia**

Configuring OpenLDAP

OpenLDAP consists of slapd and slurpd daemon. This howto covers one LDAP server without a replication, so we will focus only on slapd. I also assume you installed and initialized your openldap installation (depends on system/disribution). If so, let's go to configuration part.

On my system (Gentoo), openldap's configuration is stored in `/etc/openldap`, we are interested in `/etc/openldap/slapd.conf` file. But first we have to generate a password for LDAP administrator, to put it into the config file:

```
slappasswd -h {md5}
```

The config looks like this:

```
include         /etc/openldap/schema/core.schema

include         /etc/openldap/schema/cosine.schema

include         /etc/openldap/schema/inetorgperson.schema

include         /etc/openldap/schema/nis.schema

allow bind_v2



pidfile         /var/run/openldap/slapd.pid

argsfile        /var/run/openldap/slapd.args



modulepath      /usr/lib/openldap/openldap



access to attrs=userPassword

        by dn="uid=root,ou=People,dc=domain,dc=com" write

        by dn="cn=Manager,dc=domain,dc=com" write

        by anonymous auth

        by self write

        by * none
```

```
access to dn.base="" by * read

access to *
        by dn="cn=Manager,dc=domain,dc=com" write
        by * read

database        bdb
suffix          "dc=domain,dc=com"
rootdn          "cn=Manager,dc=domain,dc=com"
rootpw          {MD5}Tk1sMytv5ipjr+Vhcf03JQ==

directory       /var/lib/openldap-data
index   objectClass     eq
```

Remember to change suffix and paths to your needs.

These are basic options with some basic ACLs needed to change passwrods by user. If you want more functionality, please read the manual about openLDAP. Now when we have a proper config for slapd, we can start the daemon :

```
/etc/init.d/slapd start
```

Please remember to have something like that in the config file responsible for arguments passed to the slapd (the path should point to the slapd.sock):

```
OPTS="-h 'ldaps:// ldapi://%2fvar%2frun%2fopenldap%2fslapd.sock'"
```

Now we can test if openldap is running and working properly. We do not have any data yet in the directory, but we can try to bind as cn=Manager,dc=domain,dc=com. When you are asked for password, you should use the one you generated (of course the plain text version of it :):

`ldapsearch -D "cn=Manager,dc=domain,dc=com" -W`

Migrate/Add data to the directory

Now when we have a running LDAP server, we have to fill it with data, either create or migrate entries. I will show you howto migrate existing entries from regular */etc/passwd, /etc/shadow , /etc/groups*

The first step is to configure mogrationtools to your needs. The configuration file on gentoo is located in */usr/share/migrationtools/migrate_common.ph*. Generally you need to change only these:

```
$DEFAULT_BASE = "dc=domain,dc=com";

$EXTENDED_SCHEMA = 1;
```

Now you are ready to migrate the data (actually it works even without the export command):

```
export ETC_SHADOW=/etc/shadow

./migrate_base.pl > /tmp/base.ldif

./migrate_group.pl /etc/group /tmp/group.ldif

./migrate_hosts.pl /etc/hosts /tmp/hosts.ldif

./migrate_passwd.pl /etc/passwd /tmp/passwd.ldif
```

Now we have the data in the format understood by LDAP server. Please open one the files with text editor to get used to the syntax. After that we can add the data from ldifs.

```
ldapadd -D "cn=Manager,dc=domain,dc=com" -W -f /tmp/base.ldif
```

```
ldapadd -D "cn=Manager,dc=domain,dc=com" -W -f /tmp/group.ldif
```

```
ldapadd -D "cn=Manager,dc=domain,dc=com" -W -f /tmp/passwd.ldif
```

```
ldapadd -D "cn=Manager,dc=domain,dc=com" -W -f /tmp/hosts.ldif
```

You can try searching for some data:

```
ldapsearch uid=foouser
```

Client configuration

By client I mean the machine, which connects to LDAP server to get users and authorize. It can be also the machine, the ldap server runs on. In both cases we have to edit three files : */etc/ldap.conf, /etc/nsswitch.conf and /etc/pam.d/system-auth*

Let's start woth ldap.conf, the ldap's client:

```
BASE    dc=domain, dc=com

scope sub

suffix          "dc=domain,dc=com"



## when you want to change user's password by root

rootbinddn cn=Manager,dc=domain,dc=com



## there are needed when your ldap dies

timelimit 5

bind_timelimit 5



uri ldap://ldap.domain.com/

pam_password exop
```

```
ldap_version 3

pam_filter objectclass=posixAccount

pam_login_attribute uid

pam_member_attribute memberuid



nss_base_passwd ou=Computers,dc=cognifide,dc=pl

nss_base_passwd ou=People,dc=cognifide,dc=pl

nss_base_shadow ou=People,dc=cognifide,dc=pl

nss_base_group  ou=Group,dc=cognifide,dc=pl

nss_base_hosts  ou=Hosts,dc=cognifide,dc=pl
```

Now it is time for nsswitch.conf and pamAdd these to nsswitch.conf:

```
passwd: files ldap

shadow: files ldap

group:  files ldap
```

And change the system-auth (or hatever you have like login, sshd etc) to :

```
auth       required     pam_env.so

auth       sufficient   pam_unix.so likeauth nullok

auth       sufficient   pam_ldap.so use_first_pass

auth       required     pam_deny.so



account    sufficient   pam_unix.so

account    sufficient   pam_ldap.so

account    required     pam_ldap.so



password   required     pam_cracklib.so difok=2 minlen=8 dcredit=2 ocredit=2 retry=3

password   sufficient   pam_unix.so nullok md5 shadow use_authtok

password   sufficient   pam_ldap.so use_first_pass

password   required     pam_deny.so



session    required     pam_limits.so
```

```
session    required     pam_unix.so

session    optional     pam_ldap.so
```

Time to test it. The best tool for it is a good old getent. Pick a user from your system and issue:

```
getent passwd | grep foouser
```

You should get the result twice, if so the nss_ldap works fine. The pam part can be tested by deleting a user from the /etc/passwd and trying to log in through ssh.

Apache mod_auth_ldap

To have LDAP authorization in apache, you have to load mod_auth_ldap module

```
LoadModule mm_auth_ldap_module modules/mod_auth_ldap.so
```

Now it is enought to make .htaccess like that:

```
AuthName "Restricted"

AuthType Basic

AuthLDAPURL ldap://ldap.domain.com:389/ou=People,dc=domain,dc=com?uid

AuthLDAPBindDN "cn=Manager,dc=domain,dc=com"

AuthLDAPBindPassword "your_secret_secret_password_to_ldap_admin"

require valid-user
```

Note that this method can be also used for webdav subversion authorization## Administration tools for ldap

There are few tool I recommend using to administrate OpenLDAP server

- **phpldapadmin** - web based tool
- **ldapvi** - vim browsing
- **PADL migrationtools** - migrationtools
- **IDEALX sambaldap tools** - samba ldap tools## Other ldap aware applications

- Postfix
- Courier IMAP
- jabberd
- eGroupware## Summary

If someone has something to add, please do it. I know the configuration may not be perfect.